

Kvalitet softvera

U okviru ove lekcije fokusiraćemo se na sam kvalitet programa. Pre svega, definisaćemo šta je to kvalitet softvera; govorićemo o zahtevima čija ispunjenost odlikuje softver kao kvalitetan. Napravićemo i osvrt na posledice i troškove kad je reč o postizanju i održavanju kvaliteta; lekciju ćemo završiti pojmom životnog ciklusa softvera.

Sam termin *kvalitet programa* je dosta širok i može da znači mnogo toga, sve u zavisnosti od toga sa koje tačke gledišta posmatramo. Mnogi autori i organizacije tokom godina su ga različito definisali, pa pogledajmo par slučajeva:

- Phil Crosby, autor mnogobrojnih knjiga u oblasti menadžmenta i upravljanja kvalitetom, u svojoj knjizi *Philip Crosby's Reflections on Quality* definisao je kvalitet kao „usklađenost sa korisničkim zahtevima i zahtevima proizvođača”.
- IBM je dao izraz „tržišno definisan kvalitet”, što se temelji na postizanju ukupnog zadovoljstva kupca.
- Kvalitet softvera je prema ISO/IEC 25010:2011 definisan kao „sposobnost softverskog proizvoda da zadovolji iskazane potrebe pod specifičnim uslovima”.
- Prema IEEE, P730/D8 Standard for Software Quality Assurance Processes, kvalitet softvera je definisan kao „stepen do kojeg softverski proizvod ispunjava postavljene zahteve, odnosno kvalitet zavisi od toga koliko postavljene zahteve precizno predstavljaju potrebe, želje i očekivanja zainteresovanih učesnika”.

Kao što vidite, sve definicije tiču se usklađenosti sa zahtevima, odnosno ističu jaku povezanost kvaliteta sa zahtevima. Razlog ovoga je što softver uvek radimo sa namenom – stoga zahtevi nastaju planiranjem funkcije softvera, bilo da zahteve za funkcijom dobijamo od klijenta ili povratnom informacijom od korisnika itd. Stoga možemo reći da je jedan program kvalitetan ukoliko ispunjava sve zahteve koji su definisani kada je program bio u razvoju, a ujedno u svojoj primeni zadovoljava potrebe korisnika.

U okviru definicija pojma nije precizno određen tip zahteva programa koji treba da se ispunjava – funkcionalni zahtev programa, nefunkcionalni zahtev poput sigurnosti i slično. U nastavku, govorićemo upravo o zahtevima za kvalitet softvera.

Zahtevi za kvalitet softvera

Ono što je važno odmah napomenuti jeste da se termini „kvalitet softvera” i „zahtevi za kvalitet softvera” često poistovećuju. Kvalitet softvera predstavlja konačno stanje programa, dok zahtevi za kvalitetom softvera omogućavaju precizno definisanje toka i ishoda rada programa. Na taj način osiguravamo kvalitet – ukoliko se, naravno, zahtevi poštuju prilikom izrade programa. Zahtevi za kvalitet softvera su zapravo atributi ili ograničenja funkcionalnih zahteva.

Ovo možemo videti unutar vodiča SWEBOK, gde možemo videti sledeću definiciju: „Kvalitet softvera se postiže usklađenošću sa svim zahtevima, bez obzira na to koje karakteristike su definisane ili kako su zahtevi grupisani ili imenovani.” (SWEBOK, 2021, Strana 174)

Jedna od najvažnijih organizacija u oblasti softverskog inženjerstva je upravo prethodno citirani SWEBOK, pa stoga, pre nego što prikažemo osnovna načela kvaliteta prema SWEBOK-u, pogledajmo prvo šta je to tačno SWEBOK i koji je njegov značaj.

Pod okriljem IEEE Computer Society, 1997. godine počinje da radi Software Engineering

Coordinating Committee, prečišćavajući opšta znanja u oblasti softverskog inženjerstva. Cilj je bio dalje definisanje celokupne profesije softverskog inženjerstva i formiranje standarda u ovoj oblasti. Vodič SWEBOK opisuje sveukupno znanje unutar profesije softverskog inženjerstva.

Kao rezultat nastao je skup dokumenata *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*. Na kreiranju ovoga vodiča radilo je nekoliko profesionalnih tela, kao i veliki broj pojedinaca; objavilo ga je IEEE Computer Society. *SWEBOK Guide* opisuje opšte prihvaćeno znanje u oblasti softverskog inženjerstva; potpuno je besplatan i dostupan je na linku:

<https://www.computer.org/education/bodies-of-knowledge/software-engineering>

SWEBOK definiše faktore i zahteve koji utiču na postizanje kvaliteta softvera kao sledeće:

Osnovni cilj je isporuka najveće maksimalne vrednosti za korisnika (value), pritom balansirajući troškove i rokove (usefulness of something, fitness for use).

Vrednost za klijenta (stakeholder value) iskazuje se u zahtevima softvera; korisnici kod softverskih proizvoda vrednuju cenu koju su platili za softver, koliko brzo su ga dobili (isporuka) i softverski kvalitet. U slučaju vrednosti za klijenta, dogovor o kvalitetu programa dobijamo jasno definisanim funkcionalnim i nefunkcionalnim zahtevima. U tabeli 2.1. možete videti primer dela funkcionalnih zahteva za softver koji je namenjen upravljanju učenjem.

Zahtev	Opis	Prioritet
Logovanje	Sistem treba da obezbedi autentifikaciju korisnika koji mu pristupa sa adekvatnom validacijom za svaki nivo pristupa.	1
Upis polaznika	Sistem će omogućiti administratoru unos novog polaznika u evidenciju upisanih učenika.	1
Izmena rasporeda nastave	Sistem će omogućiti administratoru da kreira izmene raspored nastave.	2
Evidencija literature	Sistem će omogućiti administratoru da vodi evidenciju o školskim knjigama.	3

Tabela 2.1. Pimer funkcionalnih zahteva LMS sistema

Dakle, na ovom primeru možete videti formulisan deo funkcionalnih zahteva. Naravno, ovo je uprošćeno za potrebe primera, ali možete uočiti da se definiše tačan opis funkcionalnosti, kao i njen prioritet u razvojnom procesu. Dakle, opis bi u slučaju realnog plana za softver bio znatno detaljniji, sa tačnim uslovima kada se operacija obavlja. Da bi se softver u osnovi smatrao kvalitetnim, potrebno je da ispunjava ove uslove.

Kao što smo i ranije naveli, pored funkcionalnih zahteva, postoje i nefunkcionalni, pa bi u slučaju ovog primera programa ovi zahtevi bi mogli da budu:

Zahtev	Opis
Performanse	Aplikaciju treba postaviti na brži server koji će omogućiti bolju reakciju na korisničke akcije.
Bezbednost	U okviru sistema za upravljanje bazom podataka vrši se implementacija ograničenja i procedura koje će izvršavati proveru prilikom unosa ili izmena podataka.
Sigurnost	Podacima u sistemu škole neće moći da pristupaju svi korisnici. Zaposleni u školi imaće veća ovlašćenja od ostalih korisnika, a jedino će administrativnim radnicima biti dozvoljeno ažuriranje podataka bitnih za poslovanje škole.
Pouzdanost	Do iznenadnog prestanka rada sistema može doći jedino kao posledica grešaka u radu operativnog sistema korisnika ili problema u funkcionisanju servera.

Tabela 2.2. Primer nefunkcionalnih zahteva LMS sistema

Kvalitet softvera se posebno razmatra tokom kompletnog procesa softverskog inženjerstva i povezan je sa svim oblastima razvoja softvera, ne samo sa inicijalnim planiranjem. U nastavku, pružićemo uvid u još nekoliko usko povezanih aspekata.

Vrednost i cena kvaliteta

Definisanje i dostizanje kvaliteta softvera je kompleksan proces; često radi utvrđivanja nivoa kvaliteta softvera, npr. radi dostizanja vrednosti za zainteresovane strane, koristimo termin *cost of software quality* (CoSQ). Premisa na kojoj se temelji CoSQ je da nivo kvaliteta u softverskom proizvodu može biti izveden iz cene aktivnosti povezanih sa bavljenjem posledicama lošeg kvaliteta. Loš kvalitet znači da softverski proizvod ne zadovoljava navedene potrebe ili uspostavljene zahteve.

Naravno, postoje troškovi koji se javljaju i u slučaju da smo ispunili sva očekivanja klijenta po pitanju kvaliteta softvera, ne samo u slučaju lošeg kvaliteta. Drugim rečima, pored troškova koji se javljaju kada kreiramo defektan proizvod, imamo i troškove kada kreiramo proizvod bez defekata. Razlog pojave ovih troškova je to što u proizvod visokog kvaliteta ulažemo više vremena i truda.

Na osnovu ovoga su izvedene četiri kategorije kvaliteta u odnosu na cenu:

- **Prevenција** – Troškovi prevencije obuhvataju investicije u unapređenje softverskog procesa, infrastrukturu kvaliteta, alate kvaliteta, trening, revizije i kontrolu menadžmenta. Ovi troškovi nisu vezani samo za dati projekat, već obuhvataju celu organizaciju
- **Procena** – Troškovi procene nastaju usled aktivnosti pronalaženja defekata. Ove aktivnosti mogu da se podele na troškove pregleda i troškove testiranja. Troškovi testiranja obuhvataju testove jedinice, testove integracije, sistemsko testiranje i test prihvatljivosti – upravo vrste testiranja na kojima ćemo držati fokus u nastavku kursa. Po pitanju troškova pregleda, ovde imamo poklapanja sa samom kategorijom prevencije, prvenstveno u pogledu revizija proizvoda, kako kad je reč o samom programskom kodu tako i kad se radi o praćenju zadovoljstva korisnika, tj. klijenta, stabilnosti, bezbednosti i drugih faktora.
- **Interni otkazi** – troškovi internih otkaza su troškovi otklanjanja grešaka i defekata otkrivenih tokom aktivnosti procene, a pre isporuke proizvoda. Pod internim otkazima podrazumevamo probleme gde program ne obavlja neku traženu funkciju ili ne radi u potpunosti, ali klijent nije počeo da ga koristi ili nije obajavljen. U ovom slučaju, klijent retko snosi troškove, jer je odgovornost razvojnog tima da ostvari dogovorene ciljeve, ali izaziva troškove poput pomeranja dogovorenih rokova i izgubljenog vremena.
- **Eksterni otkazi** – Troškovi eksternih otkaza su troškovi otklanjanja grešaka i defekata otkrivenih nakon isporuke korisnicima. Ovo je jedan od najtežih načina da otkrijemo manu u radu programa. U ovom slučaju program ne radi, a isti već je objavljen i aktivno se koristi. Eksterni otkazi mogu biti izazvani hakerskim upadima, neusklađenošću rada u odnosu na broj korisnika... Ove otkaze odlikuju vrlo kratki rokovi, jer se događaju kada je program/aplikacija već u primeni. U zavisnosti od posla klijenta, troškovi mogu biti milionski, jer usled nepouzdanog softvera može doći do smanjenja baze korisnika, smanjenja prodaje, plaćanja penala i slično.

Softver inženjeri treba da koriste CoSQ metode radi utvrđivanja nivoa kvaliteta softvera i treba da prezentuju alternative po pitanju kvaliteta i njihovih troškova kako bi se napravio balans između troškova, rokova i isporuke vrednosti za naručioca. Definisanje troškova postizanja kvaliteta takođe doprinosi preciznijem definisanju budžeta i troškova održavanja softvera u inicijalnim fazama projekta.

Obezbeđenje i kontrola kvaliteta

Quality assurance (QA) je proces za prevenciju defekata i grešaka u proizvodu i izbegavanje problema pre isporuke finalnog rešenja ili proizvoda naručiocu. QA obuhvata administrativne i proceduralne aktivnosti koje se sprovode u sistemu kvaliteta kako bi se ispunili zahtevi i ciljevi za proizvod ili servis. Obuhvata sistematsko merenje, poređenje sa standardom, te praćenje procesa i pridruženih povratnih sprega koje omogućava prevenciju grešaka.

Naravno, definicija nam je manje važna, ali imamo dva ključna principa kojima se uvek vodimo kada razmišljamo o obezbeđenju kvaliteta; to su:

- *fit for purpose* – proizvod treba da bude odgovarajući za datu namenu i
- *right first time* – greške treba da budu izbegnute.

Iz ova dva pristupa, razvojem oblasti su nastale dve podoblasti: *quality assurance* i *quality control*; ovi termini se danas često koriste kao sinonimi kako bi se osigurao kvalitet servisa ili proizvoda.

Važno je znati da su ovo različite oblasti, gde je *quality assurance* proces fokusiran na prevenciju defekata (*right first time*), dok *quality control* predstavlja proces fokusiran na identifikaciju defekata koji su se javili tokom razvoja ili primene proizvoda (*fit for purpose*).

Upotreba ovih termina kao sinonima ima i svoju dobru stranu, gde ove dve odvojene oblasti postaju jedna i na ovaj način prevencija i identifikacija postaju deo jedne celine postizanja kvaliteta – ukoliko, naravno, razmišljamo i koristimo pristupe na ovaj način.

Pitanje

Utvrdite tačnost iskaza.

Eksterni otkazi predstavljaju greške i defekte koji su otkriveni nakon isporuke korisnicima.

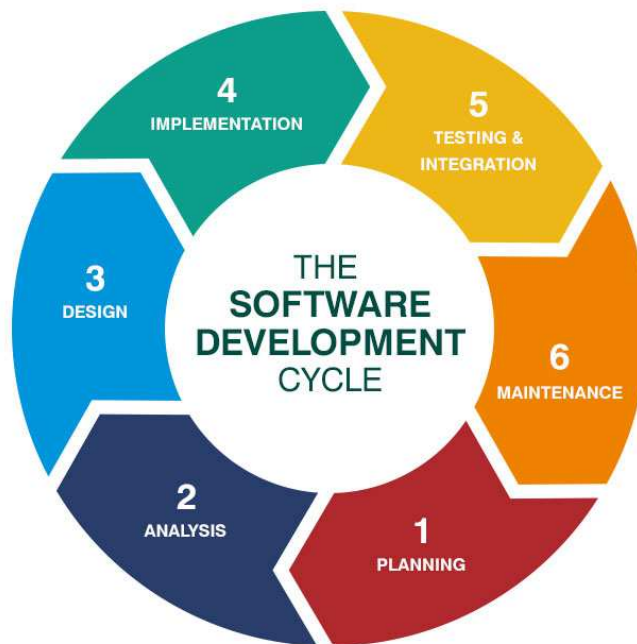
- **Tačno.**
- Netačno.

Objašnjenje:

Eksterni otkazi podrazumevaju otklanjanje grešaka i defekata otkrivenih nakon isporuke korisnicima. Ovo je jedan od najtežih načina otkaza programa, jer u ovom slučaju manu otkrivamo kada je program već u primeni i korisnici ga aktivno upotrebljavaju. Eksterni otkazi mogu biti izazvani hakerskim upadima ili neusklađenošću rada u odnosu na broj korisnika.

Životni ciklus razvoja softvera

Pre svega – šta je to životni ciklus? Ukoliko posmatramo proces razvoja određenog proizvoda, taj proces možemo definisati kao životni ciklus proizvoda. Ako se radi o softveru, govorimo o životnom ciklusu razvoja softvera. „Život” softverskog proizvoda podrazumeva ceo ciklus, od prvih zahteva, preko dizajna, razvoja, korišćenja i održavanja, pa sve do trenutka povlačenja softvera iz upotrebe. Na slici 2.1. možete videti primer dijagrama životnog ciklusa softvera.



Slika 2.1. Ilustracija razvoja softvera¹

Kao što možete videti, u dosadašnjem radu smo prošli faze planiranja zahteva, analize zahteva, samog dizajniranja i implementacije i došli smo do pete faze, koja je i centralna tema ovog kruga. Ove uvodne lekcije koristimo za upoznavanje sa teorijom i osnovnim postulatima softverskog inženjerstva, a ujedno i obezbeđenja kvaliteta, koji su nam potrebni za razumevanje potrebe za testiranjem programa i razvojem kvalitetnih testova koji će omogućiti proveru našeg softvera u odnosu na zahteve klijenata i očekivani rad u primeni softvera. U narednoj lekciji započinjemo sa detaljnijim upoznavanjem procesa testiranja programa.

Rezime

- Kvalitet softvera možemo definisati kao usklađenost softvera sa zahtevima koji su definisani kada je program bio na samom početku razvojnog procesa.
- Vrednost za klijenta se iskazuje u zahtevima softvera. Korisnici kod softverskih proizvoda vrednuju cenu koju su platili za softver, koliko brzo su ga dobili i softverski kvalitet.
- Funkcionalni zahtevi definišu tačan opis funkcionalnosti programa i njen prioritet u razvojnog procesu.
- Pored funkcionalnih zahteva, postoje i nefunkcionalni zahtevi. To su: brzina, sigurnost i stabilnost softvera.
- Cost of software quality (CoSQ) predstavlja nivo kvaliteta u softverskom proizvodu koji može biti izveden iz cene aktivnosti povezanih sa bavljenjem posledicama lošeg kvaliteta proizvoda. Pored troškova koji se javljaju kada kreiramo defektan proizvod, imamo troškove i kada kreiramo proizvod bez defekata. Razlog pojave ovih troškova je to što u proizvod visokog kvaliteta ulažemo više vremena i truda.

¹ <https://bigwater.consulting/2019/04/08/software-development-life-cycle-sdlc/>

- Na osnovu CoSQ izvedene su četiri kategorije kvaliteta u odnosu na cenu: prevencija, procena, interni otkazi i eksterni otkazi.
- Quality assurance (QA) predstavlja proces za prevenciju defekata i grešaka u proizvodu i izbegavanje problema pre isporuke finalnog rešenja ili proizvoda naručiocu. Obuhvata sistematsko merenje, poređenje sa standardom, te praćenje procesa i pridruženih povratnih sprega koje omogućavaju prevenciju grešaka.
- Quality assurance predstavlja proces u kojem je fokus na prevenciji defekata (right frist time), dok quality control predstavlja proces fokusiran na identifikaciju defekata koji su se javili tokom razvoja ili primene proizvoda (fit for purpose).
- Životni ciklus softverskog proizvoda podrazumeva ceo proces razvoja i upotrebe jednog programa, od prvih zahteva, preko dizajna, razvoja, korišćenja i održavanja, pa sve do trenutka povlačenja softvera iz upotrebe.

